Application of Multiple-Criteria Decision Analysis in Open Distributed Processing Systems Management

Romain Lenglet
lenglet@csg.is.titech.ac.jp
JSPS Postdoctoral Fellow,
Tokyo Institute of Technology

August 3, 2005

1 Conformance assessment

Conformance assessment of an Open Distributed Processing (ODP) system is the verification that the ODP system performs as expected by human stakeholders. What is expected (the *goals*) is expressed in the form of *conformance requirements*, i.e. as parts of systems specifications that can be tested. *Conformance* is a relation between a specification and a real implementation, and holds when specific conformance requirements in the specification are met by the implementation [3]. The first problem, which is addressed in the domain of *requirements engineering* (RE), is to elicite, elaborate, structure, specify, analyze, negotiate, document, and modify such conformance requirements [7]. The next problem is to ensure that such requirements hold, for which two main approaches exist, which are called *provisioning* and *outsourcing* in the domain of networks management [8].

On one hand, *provisioning*, or *analysis and design* (AD), is the consistent translation and refinement of high-level enterprise specifications of a system, that include conformance requirements, down to a system implementation [3]. In AD, assessing conformance of a system is achieved by 1) *checking refinements* between specifications, 2) *checking internal validity* of single specifications, 3) *checking consistency* of a number of specifications, and 4) *testing a realization*, between a real implementation and a specification, to confirm the claim that the implementation's behaviour or properties are as required by the specification as expressed in the conformance requirements [3]. On the other hand, *outsourcing*, or *systems management* (SM), conformance assessment is achieved by developing a *management system* (MS), that interacts with the system at runtime, to make it conformant again when its observed behaviour is no more conformant to requirements. We consider that all three activities (RE, AD and SM) must all be performed jointly in order to completely assess the conformance of a system. For instance, only the SM activity can achieve fault tolerance.

2 Management systems design

At first, we are more precisely addressing the problem of the design of MS, and more precisely of the conformance assessment of MS, i.e. we try to answer the question: how much can one be confident that a MS makes a managed system conformant to requirements? That problem has seldom been addressed in the literature: most existing works address technological issues (*what* MS are), e.g. the architecture of MS [2]. However, no existing work has addressed the problem of guiding the design of MS (*how* to design MS), by combining existing technologies, to make managed systems conformant to requirements. The solution that we propose to solve this problem is in the form of a *MS design method*.

Our proposal is original and is motivated by the idea that the conformance assessment of MS is different from that of "normal" ODP systems. Our study has lead to the definition of at least two requirements for design methods to be suitable for the design of MS[1]. Firstly, it is necessary to evaluate and express the *effects* of a MS on a managed system, to assess if that strategy makes the managed system conformant, in a *consequentialist* philosophy. However, to our knowledge, no existing method or specification language allows to specify effects of actions of a system on another system: most formalisms are based on *deontic logic*[5], i.e. on *deontology*, which may be useful for the specification of "normal" systems, but not of MS. Secondly, in order to compare the effects of management streategies (i.e. sequences of actions) on the managed system, with its conformance requirements, it is at least necessary to specify them separately. However, most existing MS approaches follow the so-called "policy-based management" approach [6], which does not distinguish strategies from requirements.

3 Applying MCDM

Our proposal is founded on the idea that the important part of SM is to *choose* management strategies that best make a managed system conformant. The

[1] Although we have identified more than 15 requirements, only 2 are introduced here due to limited space.

problem is to make such a decision in the presence of *uncertainty* about the state of the managed system, the effects of management strategies, etc. We therefore propose to apply Multiple-Criteria Decision Analysis (MCDA) methods, that have been used successfully for deciding in the real world for more than 20 years. The fundamental idea in MCDM is to model the decision problem along several dimensions, or criteria, instead of a unique criterion (e.g. money) as in classical optimization techniques, to better deal with uncertainty and conflicting goals [1]. In the context of SM, we propose that this multi-dimensional model must reflect the state of the managed system, as can be measured at conformance points (e.g. at interfaces of objects). In that model must be expressed separately both the conformance requirements to enforce (e.g. as reference points, i.e. as expected values of criteria), and the predicted effects of management strategies.

Then, we propose to describe the design of a MS as the following decision process: 1) *identify* a set of interesting management strategies, 2) *define* the criteria that form the model of the state of the managed system, 3) *evaluate the consequences* of management strategies on the managed system's state, expressed in that model, and 4) at runtime, the MS must *compare* those consequences to the conformance requirements that must be enforced, and *choose* and execute the management strategy that best makes the managed system meet the conformance requirements. Our proposal is concretely in the form of a precice *method*, i.e. as an orderly procedure, that fits this general decision process. Our method allows to use any MCDM real-world method at the *methodological* level, in order to best address the problem to solve, because it does not make any assumption other than those of the *technical* level that is common to all decision methods [4]. Some characteristics of our method are: 1) last step of decision is performed automatically by a MS, making it *autonomous*, whereas MCDM is usually only an aid to human stakeholders who make decisions, 2) our method is iterative, i.e. the decision process above is repeated until the requirements and strategies satisfy stakeholders' goals, then the MS can be implemented, 3) our method allows for multiple conflicting stakeholders' viewpoints, through a two-level mapping of requirements to conformance points (called *subjective* and *objective* mappings in the literature).

4 Conclusion

Our proposal has been validated by the development of a prototype MS in Java, using the Fractal software component model, for the management of an ODP system to enforce QoS requirements that are different between clients. Our research of the bibliography and the theoretical foundations of our work is now finished, and is the subject of another long article that is being written. The first perspective is to link our MCDM-based approach of SM, to existing MCDM-based approaches of RE and AD, in order to *unify* all three activities under the same conceptual framework (MCDM) and methods, and to describe them as similar decision problems, in order to assess more easily and efficiently the conformance of ODP systems. Another perspective is to study the problem of managing Aspect-Oriented (AO) systems to make them conformant, since the literature demonstrates that the AO design of systems makes it easier to translate requirements in practice (this approach is called "Early Aspects"), and therefore to make systems conformant to requirements.

References

[1] D. Bouyssou. Building criteria: a prerequisite for MCDA. In C. A. B. e Costa, editor, *Readings in Multiple Criteria Decision-Aid*, pages 58–80. Springer-Verlag, Nov. 1990.

[2] ISO/ITU–T. Open Distributed Management Architecture – international standard ISO/IEC 13244, ITU–T recommendation X.703, Oct. 1997.

[3] ISO/ITU–T. Reference Model of Open Distributed Processing (RM-ODP): Enterprise language – international standard ISO/IEC 15414, ITU–T recommendation X.911, Oct. 2001.

[4] I. Kaliszewski. Out of the mist – towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research*, 158(2):293–307, Oct. 2004.

[5] P. Linington, J. Derrick, and H. Bowman. The specification and testing of conformance in ODP systems. In *Proceedings of the 9th IFIP International Workshop on Testing of Communicating Systems*, pages 93–114, Darmstadt, Germany, Sept. 1996. Chapman & Hall.

[6] M. Sloman. Policy driven management for distributed systems. *Journal of Network and Systems Management*, 2(4):333–360, Dec. 1994.

[7] A. van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pages 249–262, Washington, DC, USA, 2001. IEEE Computer Society.

[8] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser. Terminology for policy-based management. RFC 3198 (Informational), Nov. 2001.